

IHM

-hConsole

HANDLE

-activerCouleur

vector<string>

-colorerEcran(int code_couleur)

bool

+choixDepartArrivee(metro : Metro)

void

+choixTypeItineraire()

bool

+afficherItineraire(itineraire : Itineraire)

void

+quitter()

bool

Itineraire

-arretsItineraire

vector<Arret*>

-tempsTotal

int

+Itineraire(destination : Node*, metro : Metro)

Itineraire

+creerItineraireSimplifie()

vector<Arret*>

+creerNbArretsSimplifie()

vector<int>

Metro

-arretsMetro

vector<Arret*>

-lignesMetro

vector<Ligne*>

+importerDonnees(wd : string)

void

+getArret(idArret : int)

Arret*

+getLigne(route_id : int)

Ligne*

-setArretsMetro(arret : Arret*)

void

-importerToutesLignesTxt(wd : string)

void

-importerTousStopTxt(wd : string)

void

-importerStopTxt(path_fichier : string)

void

-importerLigneTxt(path_fichier : string)

bool

Ligne

-idLigne

int

-numero_ligne

string

-nom_trajet

string

-couleur

string

-direction

string

-arretsLigne

vector<Arret*>

+setArretsLigne(arret : Arret*)

void

Arret

-idArret

int

-nom

string

-adresse

string

-stop_lon

float

-stop_lat

float

-lignesArret

vector<Ligne*>

+setLignesArret(ligne : Ligne*)

void

+memeLigne (arret : Arret*)

bool

+memeArret (arret : Arret*)

bool

-calculerDirection(arret : Arret*)

vector<string>

Métro

Graphe

-nodes

vector<Node*>

-edges

vector<Edge*>

-adjacentRemainingNodes(node : Node*)

vector<Node*>*

-initialiserNodes()

void

-importerNodes(chemin : string)

void

-importerEdges(chemin : string, chemin_changement : string)

void

-extractSmallest(&nodes : vector<Node*>, minChangement : bool)

Node*

-contains(&nodes : vector<Node*>, node : Node*)

bool

-calculerDistance(node1 : Node*, node2 : Node*, minChangement : bool)

int

+dijkstras(entree : string, sortie : string, minChangement : bool)

Node*

-getIndiceFromNode(identifiant : string, les_noeuds : vector<Node*>)

int

Edge

+node1

Node*

+node2

Node*

-distanceCourtChemin

int

-distanceMinChangement

int

+connects(node1 : Node*, node2 : Node*)

bool

+getDistance(minChangement : bool)

int

Node

-idNode

int

-distanceFromStartCourtChemin

int

-distanceFromStartMinChangement

int

+previous

Node*

+setDistanceFromStart(dist : int, minChangement : bool)

void

+getDistanceFromStart(minChangement : bool)

int

+initialiserNode()

void

Algorithme